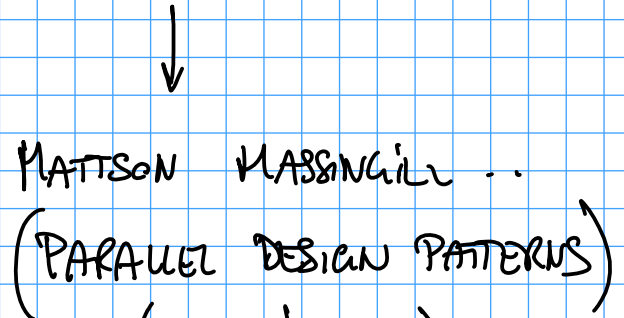
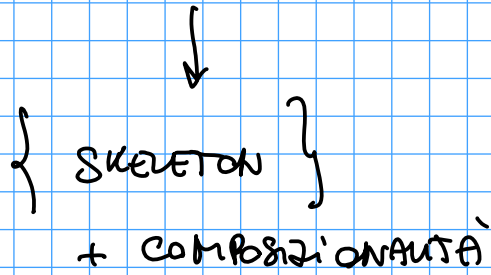


MODELLI di PROG. PAR. STRUTTURATI

ALGORITHMIC SKELETONS
MURRAY Cole '88

PAR. DESIGN PATTERNS
GAMMA Book



FRAMEWORK

di prog. par. skeletons

TBB (INTEL)

TPL (MICROSOFT)

FASTFLOW (P1)

SNETO (TOXIC)

SXCEPU (LINKOPING)

MUESLI (MUENSTER)

SKANDIUM (CHILE)
JAVA

C/C++

PAR. PATTERN / ALGORITHMIC SKELETON

SINONIMI

STREAM PARALLEL

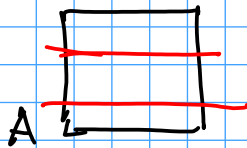
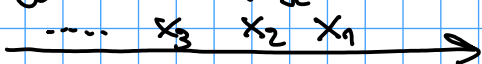
DATA PARALLEL

CONTROL PARALLEL

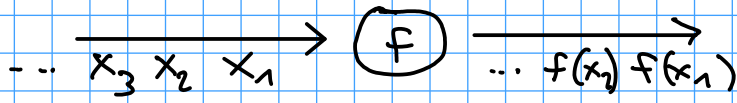
Sorgente dei dati
è uno stream (seq di
oggetti disponibili nel tempo)

Sorgente dei dati è
collezione di dati

ha un "controllo"
che determina
corte operazioni
 $f(x)$ then $\{T\}$ else $\{E\}$



STREAM PARALLEL



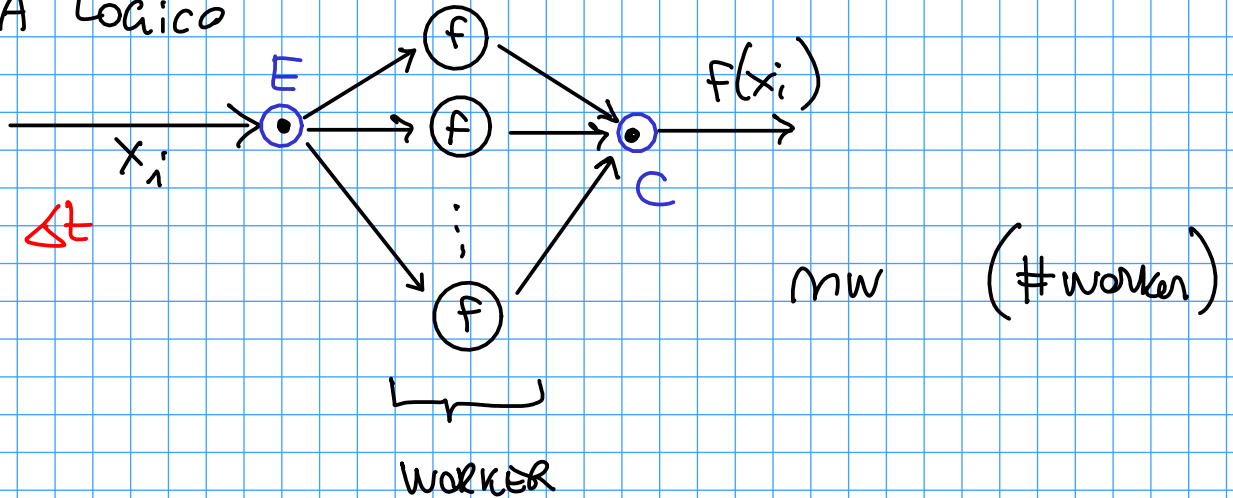
Embarrassingly
Stream
parallelism

FARM (TASK FARM, MASTER/WORKER)

f non si scompone

Unlimited + risorse per calcolare f su x_i diversi

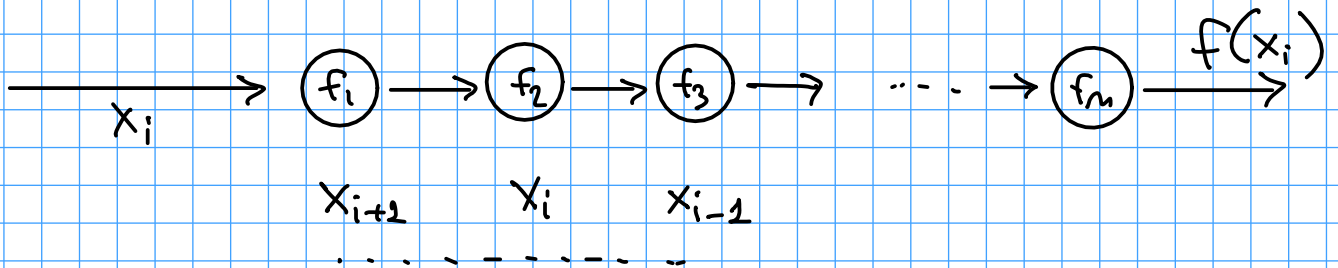
SCHEMA Logico



PIPELINE

f è scomponibile

$$f = f_1 \circ f_2 \circ \dots \circ f_m$$



FORME CHE

1) MIGLIORANO il TEMPO di SERVIZIO
smauini

2) AUMENTANO IL THROUGHPUT

DATA PARALLEL (PATTERN)

MAP on vettori

MAP: α VEC \rightarrow β -vec

$f: \alpha \rightarrow \beta$

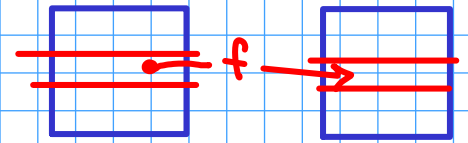
$[x_1 \dots x_n] \rightarrow \text{MAP}(f) \rightarrow [f(x_1) \dots f(x_n)]$

embarrassingly
data
parallelism

MAP su Matrici

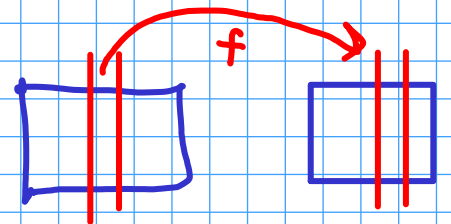
matrice :: collezione di righe

$f: \text{riga} \rightarrow \text{riga}$

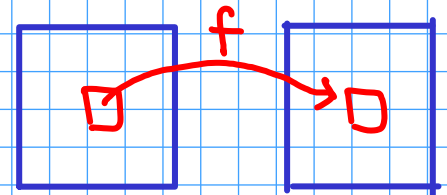


matrice :: collezione di colonne

$f: \text{col} \rightarrow \text{col}$



matrice :: collezione di elementi

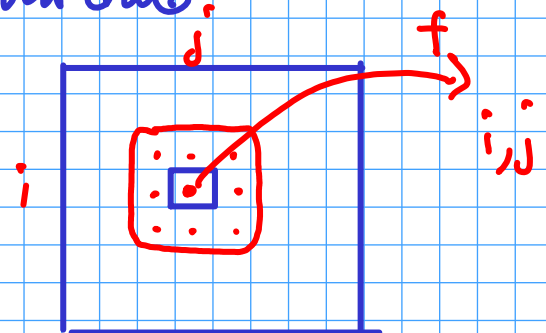


MAP: α collezione \rightarrow β collezione
 $f: \alpha \rightarrow \beta$

STENCIL

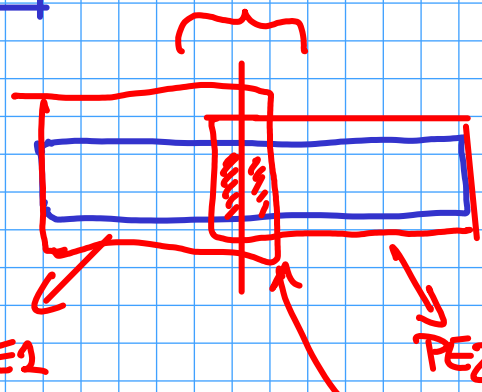
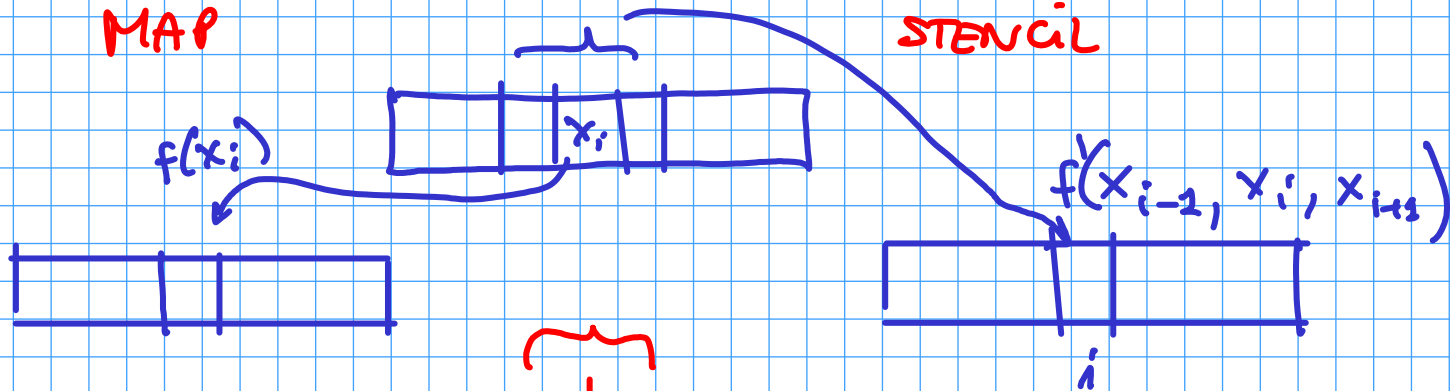
\forall elemento della collezione \rightarrow calcoliamo un "intorno"

elemento della
collezione risultante



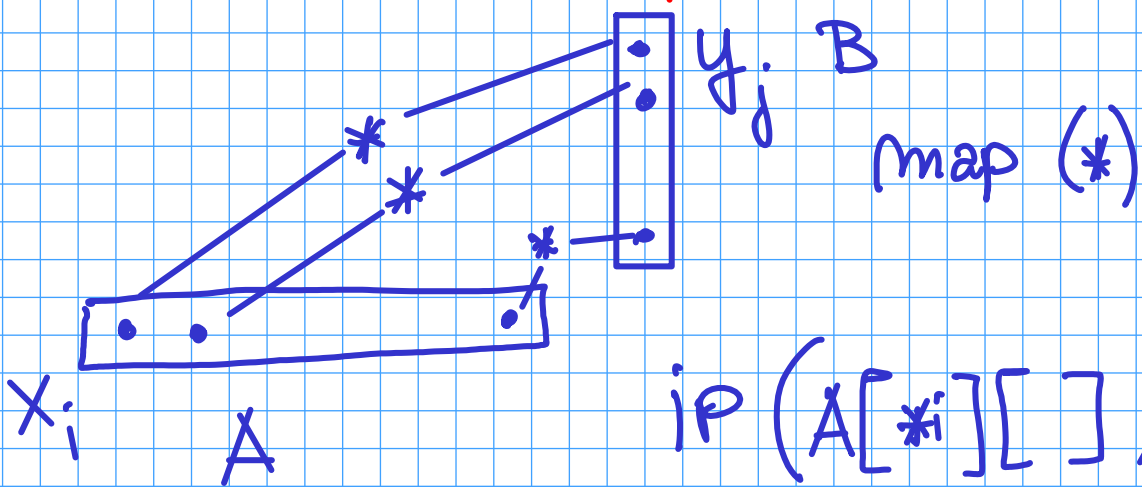
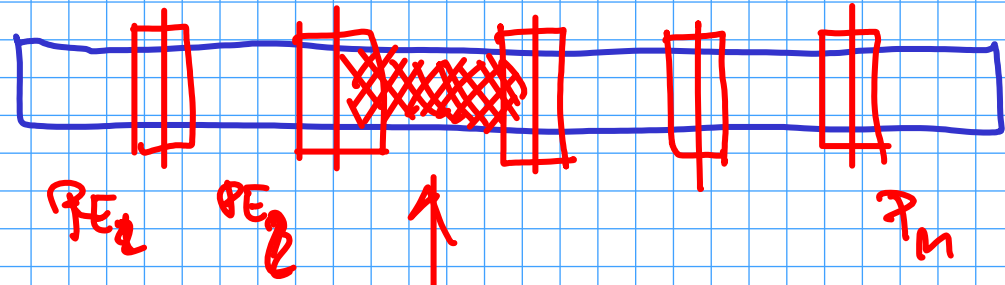
MAP

STENCIL



Scambio (PE_1, PE_2)
iterazione i

```
MAP
for(i=0; i < N/2; i++)
  y[i] = f(x[i]);
```

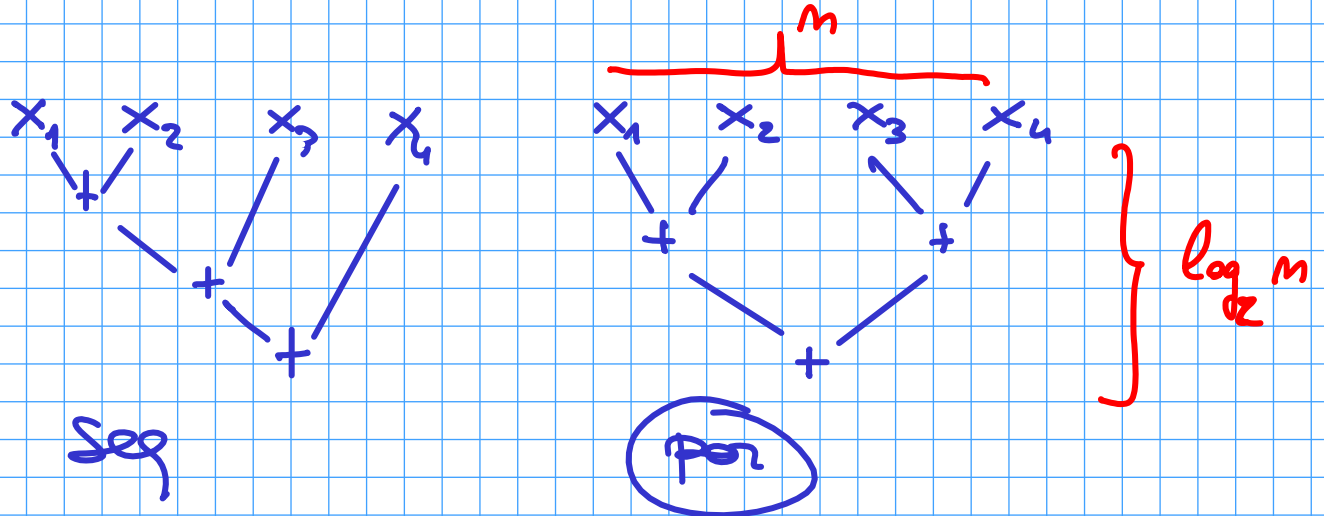


```
ip(A[*i][], B[][*j])
  a_{i0} a_{i1} a_{i2} ...
```

REDUCE

$$[x_1 \dots x_m] \xrightarrow{\text{RED}(f)} x_1 \overset{+}{f} x_2 \overset{+}{f} x_3 \dots \overset{+}{f} x_m$$

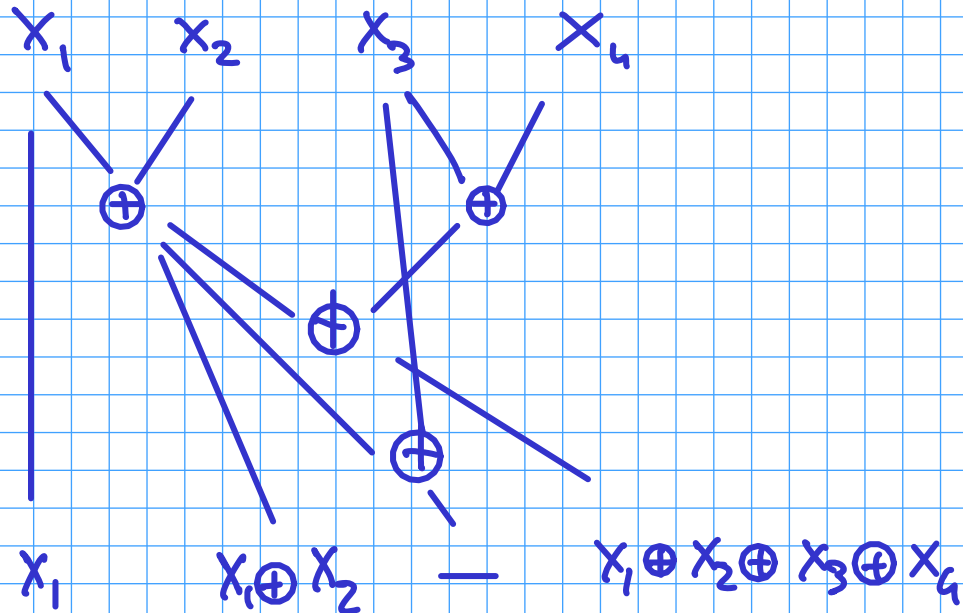
$f(\alpha \rightarrow \alpha \rightarrow \alpha)$



SCAN

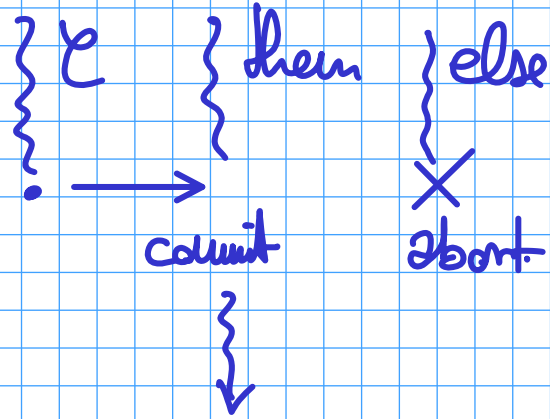
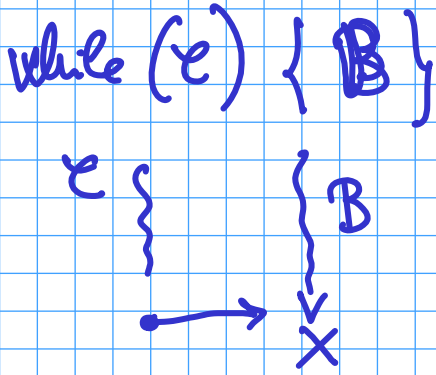
$$[x_1 \dots x_m] \rightarrow \text{SCAN}(f) \rightarrow$$

$$[x_1, x_1 \oplus x_2, x_1 \oplus x_2 \oplus x_3, \dots, x_1 \oplus x_2 \oplus \dots \oplus x_m]$$



CONTROL PARALLEL

if then else



SPECULATIVE PARALLELISM

if (x == 3)



then { a = x + y; }
else { y = x - 3; }

MUESLI

{
 pipe
 form
 map
 reduce
 stencil

SKEPU

{
 map
 reduce

Skepu:: Vector<float>

Skepu:: Map< ^{inc} _↑ > S1 = new inc()

F 1-Dim

SKEPU_UNARY_FUNC(

inc,

float,

x,

return (x+1))

Sd (x, y)

skepu: Vector